.NET FRAMEWORK SECURITY CHECKLIST

Version 1, Release 3

22 April 2016

Developed by DISA for the DoD
Trademark Information

Names, products, and services referenced within this document may be the trade names, trademarks, or service marks of their respective owners. References to commercial vendors and their products or services are provided strictly as a convenience to our users, and do not constitute or imply endorsement by DISA of any non-Federal entity, event, product, service, or enterprise.
TABLE OF CONTENTS

1. INTRODUCTION .....................................................................................................................1
   1.1 Overview ..............................................................................................................................1
   1.2 Organization of the Checklist ...............................................................................................1
   1.3 Supported Versions ..............................................................................................................2
   1.4 Review Method ....................................................................................................................2
   1.5 Referenced Documents ........................................................................................................2
   1.6 Vulnerability Severity Category Code Definitions ..............................................................2
   1.7 Document Revisions ............................................................................................................3
   1.8 Other Considerations ............................................................................................................3
   1.9 Product Approval Disclaimer ...............................................................................................3

2. .NET FRAMEWORK SRR RESULTS REPORT .................................................................5
   2.1 Site Information ...................................................................................................................5
   2.2 System Information .............................................................................................................6

3. .NET FRAMEWORK OVERVIEW .....................................................................................16
   3.1 Assemblies .........................................................................................................................16
   3.2 Evidence .............................................................................................................................16
   3.3 Permission Sets ..................................................................................................................16
   3.4 Code Group ........................................................................................................................16
   3.5 Determining Effective Permissions ....................................................................................17

4. CHECKLIST INSTRUCTIONS ...........................................................................................18
   4.1 .NET Framework Vulnerability Types ..............................................................................18
   4.2 Versions of .NET Framework Installed .............................................................................18
   4.3 Default Installations of .NET ............................................................................................18
      4.3.1 .NET Framework 1.0 Default Code Groups ............................................................20
      4.3.2 .NET Framework 1.1 Default Code Groups ............................................................21
      4.3.3 .NET framework 2.0, 3.0 & 3.5 Default Code Groups ............................................22
      4.3.4 .NET Framework 1.0 Default Permission Sets .......................................................23
      4.3.5 .NET Framework 1.1 Default Permission Sets .......................................................37
      4.3.6 .NET Framework 2.0, 3.0 & 3.5 Default Permission Sets ...........................................50
   4.4 IAVM Compliance .............................................................................................................63
   4.5 Version-specific Vulnerabilities .......................................................................................64
   4.6 .NET Configuration File Location .....................................................................................64
   4.7 Reviewing Permissions with Code Access Security Policy Tool ......................................64
   4.8 Reviewing Software Publishing State Values ....................................................................66

5. .NET SECURITY FRAMEWORK CHECKS AND PROCEDURES ...............................69
   5.1 APPNET0001: File IO Permission .................................................................................69
   5.2 APPNET0003: Isolated Storage Permission .....................................................................70
5.3 APPNET0004: User Interface Permission (Windowing) ..................................................72
5.4 APPNET0005: User Interface Permission (Clipboard) ..................................................73
5.5 APPNET0006: Reflection Permission .............................................................................74
5.6 APPNET0007: Printing Permission ...............................................................................76
5.7 APPNET0008: DNS Permission .....................................................................................77
5.8 APPNET0009: Socket Access Permission .....................................................................78
5.9 APPNET0010: Web Access Permission .........................................................................80
5.10 APPNET0011: Message Queue Permission ...................................................................81
5.11 APPNET0012: Service Controller Permission .............................................................82
5.12 APPNET0013: Database Permission ...........................................................................83
5.13 APPNET0014: Security Permission (Extend Infrastructure) .......................................85
5.14 APPNET0015: Security Permission (Enable Remoting Configuration) .......................86
5.15 APPNET0016: Security Permission (Enable Serialization Formatter) .......................87
5.16 APPNET0017: Security Permission (Enable Thread Control) ......................................89
5.17 APPNET0018: Security Permission (Allow Principal Control) ....................................90
5.18 APPNET0019: Security Permission (Enable Assembly Execution) .............................91
5.19 APPNET0020: Security Permission (Skip Verification) ...............................................92
5.20 APPNET0021: Security Permission (Allow Calls to Unmanaged Assemblies) ..........94
5.21 APPNET0022: Security Permission (Allow Policy Control) .......................................95
5.22 APPNET0023: Security Permission (Allow Domain Policy Control) .........................96
5.23 APPNET0024: Security Permission (Allow Evidence Control) ..................................97
5.24 APPNET0025: Security Permission (Assert any Permission that Has Been Granted) .99
5.25 APPNET0026: Performance Counter Permission .......................................................100
5.26 APPNET0027: Environment Variables Permission .....................................................101
5.27 APPNET0028: Event Log Permission .........................................................................102
5.28 APPNET0029: Registry Permission ..........................................................................104
5.29 APPNET0030: Directory Services Permission ............................................................105
5.30 APPNET0031: No Strong Name Verification ................................................................106
5.31 APPNET0032: First Match Code Groups .................................................................107
5.32 APPNET0033: File Code Groups, Net Code Groups ..................................................109
5.33 APPNET0035: Level Final Code Group Attribute ......................................................109
5.34 APPNET0041: Zone Membership Condition ...............................................................110
5.35 APPNET0045: Administering CAS Policy .................................................................112
5.36 APPNET0046: Software Publishing Certificate ............................................................112
5.37 APPNET0048: Publisher Membership Condition .......................................................113
5.38 APPNET0052: Strong Name Membership Condition .................................................114
5.39 APPNET0054: Administering CAS Policy for Group Names .....................................115
5.40 APPNET0055: Administering CAS Policy and Policy Configuration File Backups ....116
5.41 APPNET0060: Remoting Services Authentication and Encryption ............................116
LIST OF TABLES

Table 1-1: Referenced Documents .................................................................................................. 2
Table 1-2: Vulnerability Severity Category Code Definitions ....................................................... 3
Table 2-1: System Detail ................................................................................................................. 6
Table 2-2: Summary of .Net Framework SRR Findings By Category ........................................... 6
Table 2-3: .Net Security Framework Findings ............................................................................... 7
Table 4-1: .Net Framework Vulnerability Types .......................................................................... 18
Table 4-2: Default Installation Findings ....................................................................................... 19
Table 4-3: Location of Configuration Files .................................................................................. 64
Table 4-4: Software Publishing State Value Table ....................................................................... 68
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4-1</td>
<td>Review Software Publishing State Values</td>
<td>67</td>
</tr>
<tr>
<td>Figure 5-1</td>
<td>Sample Finding Example</td>
<td>107</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1 Overview

The .NET Framework Security Readiness Review (SRR) targets conditions that undermine the integrity of security, contribute to inefficient security operations and administration, or may lead to interruption of production operations. Additionally, the review ensures the site has properly installed and implemented the .NET environment and that it is being managed in a way that is secure, efficient, and effective. The items reviewed are based on the NSA guide, *Guide to Microsoft .NET Framework Security*. The results of the review should be recorded in the SRR Results section with the following status designations: F- Finding, N/F- Not A Finding, N/A- Not Applicable, MR -Manual Review, or NR – Not Reviewed.

DISA has assigned a level of urgency to each finding based on Chief Information Officer (CIO) established criteria for certification and accreditation. All findings are based on regulations and guidelines. All findings require correction by the host organization. Category I findings are any vulnerabilities that provide an attacker immediate access into a machine, super user access, or access that bypasses a firewall. Category II findings are any vulnerabilities that provide information that has a high potential of giving access to an intruder. Category III findings are any vulnerabilities that provide information that potentially could lead to compromise.

Note: Security patches required by the DOD IAVM process are reviewed during an operating system security review. Information for security patch compliance is available in Appendix A of this .Net Framework Security Checklist.

1.2 Organization of the Checklist

The .NET Framework Security Checklist is composed of five major sections and three appendices. The organizational breakdown proceeds as follows:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 1</td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>This section contains summary information about the sections and appendices that comprise the .NET Framework Security Checklist, and defines its scope. Supporting documents consulted are listed in this section.</td>
</tr>
<tr>
<td>Section 2</td>
<td>.NET SRR Result Report</td>
</tr>
<tr>
<td></td>
<td>This section is the matrix that allows the reviewer to document vulnerabilities discovered during the SRR process. This section is used for a .NET Framework Security review.</td>
</tr>
<tr>
<td>Section 3</td>
<td>.NET Framework Overview</td>
</tr>
<tr>
<td></td>
<td>This section describes the components of the .Net Framework architecture</td>
</tr>
</tbody>
</table>
Section 4 Checklist Instructions

This section gives reviewers more detailed information about conducting the review. Sample configuration information and default configurations information are provided. Instructions specific to conducting reviews on default installations of the .Net Framework.

Section 5 .NET Checklist Procedures

This section documents the procedures that instruct the reviewer on how to perform an SRR using the manual procedures, and how to interpret the resulting information for vulnerabilities. Each procedure maps to a specific vulnerability listed in Section 2.

1.3 Supported Versions

The vulnerabilities discussed in Section 5 of this document are applicable to .NET Framework 1.0, 1.1, 2.0, 3.0 and 3.5.

1.4 Review Method

To perform a successful Security Readiness Review (SRR), a manual process must be employed. There are currently no automated tools to check for compliance with this checklist.

Since each version of the .NET Framework is configured separately a SRR must be performed against each version of the .NET framework that is installed on the system. Refer to Section 4.2 of this document for instructions on determining which versions of the .NET Framework are installed on the system.

1.5 Referenced Documents

The following table enumerates the documents and resources consulted:

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Description</th>
</tr>
</thead>
</table>

1.6 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Category Code of CAT I, II, or III.
Table 1-2: Vulnerability Severity Category Code Definitions

<table>
<thead>
<tr>
<th>CAT I</th>
<th>DISA Category Code Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT II</td>
<td>Any vulnerability, the exploitation of which will, <strong>directly and immediately</strong> result in</td>
</tr>
<tr>
<td></td>
<td>loss of Confidentiality, Availability, or Integrity.</td>
</tr>
<tr>
<td>CAT III</td>
<td>Any vulnerability, the existence of which <strong>degrades measures</strong> to protect against loss of</td>
</tr>
<tr>
<td></td>
<td>Confidentiality, Availability, or Integrity.</td>
</tr>
</tbody>
</table>

1.7 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: disa.stig_spt@mail.mil. DISA will coordinate all change requests with the relevant DoD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA maintenance release schedule.

1.8 Other Considerations

DISA accepts no liability for the consequences of applying specific configuration settings made on the basis of the SRGs/STIGs. It must be noted that the configurations settings specified should be evaluated in a local, representative test environment before implementation in a production environment, especially within large user populations. The extensive variety of environments makes it impossible to test these configuration settings for all potential software configurations.

For some production environments, failure to test before implementation may lead to a loss of required functionality. Evaluating the risks and benefits to a system’s particular circumstances and requirements is the system owner's responsibility. The evaluated risks resulting from not applying specified configuration settings must be approved by the responsible Authorizing Official. Furthermore, DISA implies no warranty that the application of all specified configurations will make a system 100% secure.

Security guidance is provided for the Department of Defense. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied both at the device hardening level as well as the architectural level due to the fact that some of the settings may not be able to be configured in environments outside the DoD architecture.

1.9 Product Approval Disclaimer

The existence of a STIG does not equate to DoD approval for the procurement or use of a product.
STIGs provide configurable operational security guidance for products being used by the DoD. STIGs, along with vendor confidential documentation, also provide a basis for assessing compliance with Cybersecurity controls/control enhancements which supports system Assessment and Authorization (A&A) under the DoD Risk Management Framework (RMF). DoD Authorizing Officials (AOs) may request available vendor confidential documentation for a product that has a STIG for product evaluation and RMF purposes from disa.stig_spt@mail.mil. This documentation is not published for general access to protect vendor's proprietary information.

AOs have the purview to determine product use/approval IAW DoD policy and through RMF risk acceptance. Inputs into acquisition or pre-acquisition product selection include such processes as:

- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (http://csrc.nist.gov/groups/STM/cmvp/) IAW Federal/DoD mandated standards
- DoD Unified Capabilities (UC) Approved Products List (APL) (http://www.disa.mil/network-services/ucco) IAW DoDI 8100.04
UNCLASSIFIED

2. .NET FRAMEWORK SRR RESULTS REPORT

Unclassified UNTIL FILLED IN
CIRCLE ONE
FOR OFFICIAL USE ONLY (mark each page)
CONFIDENTIAL and SECRET (mark each page and each finding)

Classification is based on classification of system reviewed:
Unclassified System = FOUO Checklist
Confidential System = CONFIDENTIAL Checklist
Secret System = SECRET Checklist
Top Secret System = SECRET Checklist

This checklist becomes effective September 30, 2005

Reviewer: ______________________ Date: ______________________
Type of Review (Remote, Sample, Full):

<table>
<thead>
<tr>
<th>Finding</th>
<th>Total</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category I:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category II:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category III:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.1 Site Information

Site: ___________________________________________________________
System Administrator Information:
Name: _________________________________________________________
E-mail Address: _______________________________________________
Phone # (Commercial): ( ) DSN: ________________________________

ISSO Information:
Name: _________________________________________________________
E-Mail Address: _______________________________
Phone # (Commercial) ______________________ DSN: _____________________
2.2 System Information

Table 2-1: System Detail

<table>
<thead>
<tr>
<th>System Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>System ID or Host Name</td>
</tr>
<tr>
<td>Hardware Platform</td>
</tr>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>Operating System Version</td>
</tr>
<tr>
<td>MAC Level</td>
</tr>
<tr>
<td>Confidentiality Level</td>
</tr>
</tbody>
</table>

Table 2-2: Summary of .Net Framework SRR Findings By Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Total Possible Findings</th>
<th>Actual Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category I</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Category II</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Category III</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Total Findings</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2-3: .Net Security Framework Findings

(A=Completely Automated, MR = Partially Automated (Manual Review), NC=Can Be Automated, NR = Not Reviewed (Cannot be Automated)).

<table>
<thead>
<tr>
<th>Procedure Section #</th>
<th>Finding Information</th>
<th>Vulnerability Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Finding Details</td>
<td>SDID/ Vulnerability Key</td>
</tr>
<tr>
<td>Section #</td>
<td>Status</td>
<td>Brief Description</td>
</tr>
<tr>
<td>5.1</td>
<td>□ Finding</td>
<td>□ Not a Finding</td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td>□ Not Reviewed</td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>□ Finding</td>
<td>□ Not a Finding</td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td>□ Not Reviewed</td>
</tr>
<tr>
<td>5.3</td>
<td>□ Finding</td>
<td>□ Not a Finding</td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td>□ Not Reviewed</td>
</tr>
<tr>
<td>5.4</td>
<td>□ Finding</td>
<td>□ Not a Finding</td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td>□ Not Reviewed</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>Status</td>
<td>Finding Details</td>
</tr>
<tr>
<td>5.5</td>
<td>Finding</td>
<td>The Reflection permission is granted with unrestricted=&quot;true&quot;. The Reflection permission is granted with Flags=&quot;Member&quot;. The Reflection permission is granted with Flags=&quot;Type&quot; to software that is not a confirmed engineering tool or software interoperability service.</td>
</tr>
<tr>
<td>5.6</td>
<td>Finding</td>
<td>The Printing permission is granted with unrestricted=&quot;true&quot;. The Printing permission is granted with Level=AllPrinting.</td>
</tr>
<tr>
<td>5.7</td>
<td>Finding</td>
<td>The DNS permission is granted with unrestricted=&quot;true&quot; to assemblies that do not originate within the local network.</td>
</tr>
<tr>
<td>5.8</td>
<td>Finding</td>
<td>The Socket permission is granted with unrestricted=&quot;true&quot;. The Socket permission is granted to code that does not provide networking service. The Socket permission is granted to code that originates from an external network.</td>
</tr>
<tr>
<td>5.9</td>
<td>Finding</td>
<td>The Web permission is granted with unrestricted=&quot;true&quot;. The Web permission is granted to specific URLs that are not documented and approved for sharing data.</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><strong>Finding Information</strong></td>
<td><strong>Vulnerability Information</strong></td>
<td><strong>SDID/Vulnerability Key</strong></td>
</tr>
<tr>
<td><strong>Section #</strong></td>
<td><strong>Status</strong></td>
<td><strong>Finding Details</strong></td>
</tr>
<tr>
<td>5.10</td>
<td>Finding</td>
<td>The Message Queue permission is granted with unrestricted=&quot;true&quot;. The Message Queue permission is granted with access=Administrator to a queue that is not used by a confirmed administrative tool. The Message Queue permission is granted with access=Browse to all queues code that is not a verified administrative tool.</td>
</tr>
<tr>
<td>5.11</td>
<td>Finding</td>
<td>The Service Controller permission is granted with unrestricted=&quot;true&quot;. The Service Controller permission is granted to a service whose code is not documented as having the same level of trust and value as the service(s) itself.</td>
</tr>
<tr>
<td>5.12</td>
<td>Finding</td>
<td>The database client permission is granted with unrestricted=&quot;true&quot;. The database permission to specific providers is granted to unauthorized code.</td>
</tr>
<tr>
<td>5.13</td>
<td>Finding</td>
<td>The Security permission is granted with unrestricted=&quot;true&quot;. The Security permission is granted with Flags=&quot;Infrastructure&quot; (Extend infrastructure) to any code that has not been verified as having complete control over message processing.</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>Finding Details</td>
<td>SDID/Vulnerability Key</td>
</tr>
<tr>
<td>5.14</td>
<td>□ Finding</td>
<td>APPNET0015</td>
</tr>
<tr>
<td></td>
<td>□ Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Security permission is granted with Flags=&quot;RemotingConfiguration&quot; to code that is not highly trusted (has a strong name with public key associated with a local entity) and that does not require network access.</td>
<td></td>
</tr>
<tr>
<td>5.15</td>
<td>□ Finding</td>
<td>APPNET0016</td>
</tr>
<tr>
<td></td>
<td>□ Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Security permission is granted with Flags=&quot;SerializationFormatter&quot; to code that is not authorized as an extension to the CLR’s trusted library base.</td>
<td></td>
</tr>
<tr>
<td>5.16</td>
<td>□ Finding</td>
<td>APPNET0017</td>
</tr>
<tr>
<td></td>
<td>□ Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Security permission is granted with Flags=&quot;ControlThread&quot; to code that is not Fully Trusted (a member of the FullTrust Permission set).</td>
<td></td>
</tr>
<tr>
<td>5.17</td>
<td>□ Finding</td>
<td>APPNET0018</td>
</tr>
<tr>
<td></td>
<td>□ Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Security permission is granted with Flags=&quot;ControlPrincipal&quot; (Allow principal control) to code that is not documented as being as trusted as the most privileged system user account).</td>
<td></td>
</tr>
<tr>
<td>5.18</td>
<td>□ Finding</td>
<td>APPNET0019</td>
</tr>
<tr>
<td></td>
<td>□ Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Security permission is granted with Flags=&quot;Execution&quot; (Allow assembly execution) to unauthorized code.</td>
<td></td>
</tr>
<tr>
<td>5.19</td>
<td>□ Finding</td>
<td>APPNET0020</td>
</tr>
<tr>
<td></td>
<td>□ Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Not Reviewed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Security permission is granted with Flags=&quot;Skipverification&quot; (Skip verification) to code that is not highly trusted (has a strong name with a public key associated with a trusted party).</td>
<td></td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Section #</td>
</tr>
<tr>
<td>5.20</td>
<td>Finding Information</td>
<td>5.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.23</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Section #</td>
<td>Status</td>
<td>Finding Details</td>
</tr>
<tr>
<td>5.24</td>
<td>□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed</td>
<td>The Security permission is granted with Flags=&quot;Assertion&quot; (Assert any Permission that has been granted) to code that is not a highly trusted extension to the CLR base libraries.</td>
</tr>
<tr>
<td>5.25</td>
<td>□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed</td>
<td>The Performance Counter permission is granted with unrestricted=&quot;true&quot; to non-default permission set. The Performance Counter permission is granted to an unauthorized machine or category. The Performance Counter permission is granted with Instrument or Administrator access to code that does not provide or administer monitoring.</td>
</tr>
<tr>
<td>5.26</td>
<td>□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed</td>
<td>The Environment Variables permission is granted with unrestricted=&quot;true&quot; to a non-default permission set.</td>
</tr>
<tr>
<td>5.27</td>
<td>□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed</td>
<td>The EventLog permission is granted with unrestricted=&quot;true&quot; to a non-default permission set assigned to code that is not used to monitor system and application events.</td>
</tr>
<tr>
<td>5.28</td>
<td>□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed</td>
<td>The Registry permission is granted with unrestricted=&quot;true&quot; to a non-default permission set to Registry permissions are granted to unauthorized code.</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>Finding Details</td>
<td>SDID/Vulnerability Key</td>
</tr>
<tr>
<td>Section #</td>
<td>Status</td>
<td>Brief Description</td>
</tr>
<tr>
<td>5.29</td>
<td>Finding</td>
<td>The Directory Services permission is granted with unrestricted=&quot;true&quot; (Grant assemblies unrestricted access to all directory service paths is selected) to a non-default permission set. The Directory Services permission specifies unauthorized service paths. Write access to the Windows system directory is granted to code that is not a trusted administrative tool Browse access to the Windows system directory services (Active Directory/Global catalog IIS Metabase) is granted to code that is not of local origin (with a strong name with a public key associated with a local entity).</td>
</tr>
<tr>
<td>5.30</td>
<td>Finding</td>
<td>Strong names are simulated on a production system.</td>
</tr>
<tr>
<td>5.31</td>
<td>Finding</td>
<td>Non-default First Match Code Groups are defined.</td>
</tr>
<tr>
<td>5.32</td>
<td>Finding</td>
<td>File Code Groups have been manually added to a .config file.</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>Section #</td>
<td>Status</td>
</tr>
<tr>
<td></td>
<td>5.33</td>
<td>Finding</td>
</tr>
<tr>
<td></td>
<td>5.34</td>
<td>Finding</td>
</tr>
<tr>
<td></td>
<td>5.35</td>
<td>Finding</td>
</tr>
<tr>
<td></td>
<td>5.36</td>
<td>Finding</td>
</tr>
<tr>
<td></td>
<td>5.37</td>
<td>Finding</td>
</tr>
<tr>
<td></td>
<td>5.38</td>
<td>Finding</td>
</tr>
<tr>
<td></td>
<td>5.39</td>
<td>Finding</td>
</tr>
<tr>
<td>Procedure Section #</td>
<td>Finding Information</td>
<td>Vulnerability Information</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td></td>
<td>Status</td>
<td>Finding Details</td>
</tr>
<tr>
<td>5.40</td>
<td>Finding</td>
<td>CAS policy and configuration files are not included as part of a reliable backup strategy.</td>
</tr>
<tr>
<td></td>
<td>Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Reviewed</td>
<td></td>
</tr>
<tr>
<td>5.41</td>
<td>Finding</td>
<td>Applications assigned the <code>typefilterlevel=&quot;Full&quot;</code> attribute do not require authentication and encryption.</td>
</tr>
<tr>
<td></td>
<td>Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Reviewed</td>
<td></td>
</tr>
<tr>
<td>5.42</td>
<td>Finding</td>
<td>Verify the installed .Net Frameworks are still supported by Microsoft.</td>
</tr>
<tr>
<td></td>
<td>Not a Finding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Reviewed</td>
<td></td>
</tr>
</tbody>
</table>
3. **.NET FRAMEWORK OVERVIEW**

Programs written for the .NET Framework execute with the credentials of the user account used to launch the program. As such these programs are constrained by any operating system security settings that may be in place. This is identical to the operation of any non .NET application. The additional restrictions that may be imposed through the .NET Framework are designed to further restrict .NET applications, providing an additional layer of protection.

The following components are used to establish which permissions are granted by the .NET Framework.

3.1 **Assemblies**

An assembly is the .NET Framework's term for a program. An assembly may consist of multiple executables, DLLs, and libraries.

3.2 **Evidence**

Evidence is information about an assembly. Evidence may be contained in the assembly itself or may be presented by the host. There are currently seven types of evidence in the .NET Framework. These evidence types are:

- Application Directory – the directory where the assembly resides.
- Hash – a cryptographic hash of the assembly.
- Publisher – The publisher of the application, based upon Authenticode signing of the assembly.
- Site – The site where the assembly originated. This is only valid when the assembly is executed directly from the site.
- StrongName – A cryptographic signing of the assembly.
- URL – The URL where the assembly originated. This is only valid when the assembly is executed directly from the URL.
- Zone – The Internet Explorer Security Zone associated with the site of origin for the assembly.

3.3 **Permission Sets**

Permission Sets are groups of permission that can be granted to .NET Assemblies. There are several default Permission Sets, and non-default Permission Sets may be created.

3.4 **Code Group**

A code group is used to assign a Permission Set to an Assembly. Assemblies are placed into 1 or more Code Groups based upon the Evidence they present. As part of membership detection any membership conditions for parent code groups must also be met.

When performing an SRR of the .NET Framework it is not enough to simply evaluate the
permissions assigned to a Permission Set to determine whether a vulnerability exists or not. Code Groups which are granted that permission set must be considered as part of the evaluation process to ensure that potentially dangerous permissions are not granted to unapproved assemblies.

For example: Access to the file system is one of the permissions that can be granted through a Permission Set. These permissions range from no access to the file system, to limited access to specific files or directories, to full access to the file system. A Permission Set that grants unrestricted access to the file system is not a vulnerability in and of itself. However if that Permission Set were granted to a Code Group whose membership condition was the Internet Zone, essentially granting full file system access to any program downloaded from the Internet, then this would be a vulnerability. The same Permission Set assigned to a Code Group whose membership is restricted by a Strong Name signed assembly, where the keys used for the signing are controlled by the site, would not be considered a vulnerability.

Note: In the example above, an Assembly that is granted unrestricted access to the file system would still be restricted by the File ACLS of the system.

### 3.5 Determining Effective Permissions

.NET Framework security policies can be defined at four levels: Enterprise, Machine, User, and Application Domain. Of these four, only the Enterprise, Machine, and User levels will be considered as part of the evaluation process. The configuration information for each level is stored in configuration files within each .NET Framework directory. There is currently no central management capability, so these files must be copied to every system in order for them to be effective.

Determining the effective permissions for a given assembly involves determining which code groups the assemblies belong to and combining all of the permission sets granted to the assembly to arrive at an effective permission set. Please refer to the *NSA Guide to Microsoft .NET Framework Security*, pages 66 – 71 for a detailed description of the rules and procedures used to determine the effective permission set.
4. CHECKLIST INSTRUCTIONS

This section details the procedures needed to perform a Security Readiness Review (SRR) of a .NET Framework installation. The .NET SRR is a manual process that uses the following tools: Microsoft .NET Framework Configuration Tools Code Access Security (CAS) Policy Tool caspol.exe, regedit.exe. These tools reside in the installations directory of .Net Framework.

4.1 .NET Framework Vulnerability Types

The checklist is categorized with eleven different types of the vulnerabilities.

<table>
<thead>
<tr>
<th>Permission Vulnerabilities</th>
<th>APPNET0001-APPNET0030</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong Name Verification</td>
<td>APPNET0031</td>
</tr>
<tr>
<td>Special Code Group Type</td>
<td>APPNET0032, APPNET0033, APPNET0035</td>
</tr>
<tr>
<td>Membership Conditions</td>
<td>APPNET0041, APPNET0048, APPNET0052</td>
</tr>
<tr>
<td>CAS Policy</td>
<td>APPNET0045</td>
</tr>
<tr>
<td>Software Publishing Certificate</td>
<td>APPNET0046</td>
</tr>
<tr>
<td>Duplicate Code Group Name</td>
<td>APPNET0054</td>
</tr>
<tr>
<td>Backup Vulnerabilities</td>
<td>APPNET0055</td>
</tr>
<tr>
<td>Remoting Services and Authentication</td>
<td>APPNET0060</td>
</tr>
<tr>
<td>Unsupported .Net Framework Versions</td>
<td>APPNET0061</td>
</tr>
</tbody>
</table>

4.2 Versions of .NET Framework Installed

To determine which versions of .Net Framework are installed, search for all the Mscorlib.dll files in %systemroot%\Microsoft.NET\Framework. Click on each of the files and view properties and click Version tab to determine the version installed. If there is no Mscorlib.dll, there is no installed version of .Net Framework in that directory.

More specific information on determining versions of .Net Framework installed can be found at the following link. [http://support.microsoft.com/kb/318785](http://support.microsoft.com/kb/318785)

4.3 Default Installations of .NET

Most systems are not setup to perform .Net development. As a result most systems are installed with a default configuration of .Net Frameworks. A default installation of .Net is default configuration provided either when the operating system was first loaded or when the .Net Framework was loaded.
Note: Window 2003 has a default installation of .Net Framework 1.1 which cannot be removed.

- **Default Code Group**: Code group that are installed by the default installations of .Net 1.0, 1.1, 2.0, 3.0, and 3.5.
- **Non-Default Code Group**: Code group that is not part of the default installation of .Net.
- **Default Permission Set**: Permission set that is part of the default installation of .Net.
- **Non-Default Permission Set**: Permission set that is not part of the default installation of .Net.

If the system has a default installation of .Net then verify that default code groups and default permission sets against the list of default code groups and permission set in the following sections.

After verifying the only default code groups and permissions sets exist, use the table below to mark the findings in for a default installation.

**Table 4-2: Default Installation Findings**

<table>
<thead>
<tr>
<th>Vulnerability ID</th>
<th>Finding Status</th>
<th>Brief Description</th>
<th>CAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPNET0001</td>
<td>Not a Finding</td>
<td>File IO Permission</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0003</td>
<td>Not a Finding</td>
<td>Isolated Storage Permission</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0004</td>
<td>Not a Finding</td>
<td>User Interface Permission (Windowing)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0005</td>
<td>Not a Finding</td>
<td>User Interface Permission (Clipboard)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0006</td>
<td>Not a Finding</td>
<td>Reflection Permission</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0007</td>
<td>Not a Finding</td>
<td>Printing Permission</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0008</td>
<td>Not a Finding</td>
<td>DNS Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0009</td>
<td>Not a Finding</td>
<td>Socket Access Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0010</td>
<td>Not a Finding</td>
<td>Web Access Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0011</td>
<td>Not a Finding</td>
<td>Message Queue Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0012</td>
<td>Not a Finding</td>
<td>Service Controller Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0013</td>
<td>Not a Finding</td>
<td>Database Permission</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0014</td>
<td>Not a Finding</td>
<td>Security Permission (Extend Infrastructure)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0015</td>
<td>Not a Finding</td>
<td>Security Permission (Enable Remoting Configuration)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0016</td>
<td>Not a Finding</td>
<td>Security Permission (Enable Serialization Formatter)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0017</td>
<td>Not a Finding</td>
<td>Security Permission (Enable Thread Control)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0018</td>
<td>Not a Finding</td>
<td>Security Permission (Allow Principal Control)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0019</td>
<td>Not a Finding</td>
<td>Security Permission (Enable Assembly Execution)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0020</td>
<td>Not a Finding</td>
<td>Security Permission (Skip Verification)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0021</td>
<td>Not a Finding</td>
<td>Security Permission (Allow Calls to Unmanaged Assemblies)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0022</td>
<td>Not a Finding</td>
<td>Security Permission (Allow Policy Control)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0023</td>
<td>Not a Finding</td>
<td>Security Permission (Allow Domain Policy Control)</td>
<td>CAT II</td>
</tr>
<tr>
<td>Vulnerability ID</td>
<td>Finding Status</td>
<td>Brief Description</td>
<td>CAT</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>APPNET0024</td>
<td>Not a Finding</td>
<td>Security Permission (Allow Evidence Control)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0025</td>
<td>Not a Finding</td>
<td>Security Permission (Assert any Permission that Has Been Granted)</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0026</td>
<td>Not a Finding</td>
<td>Performance Counter Permission</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0027</td>
<td>Not a Finding</td>
<td>Environment Variables Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0028</td>
<td>Not a Finding</td>
<td>Event Log Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0029</td>
<td>Not a Finding</td>
<td>Registry Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0030</td>
<td>Not a Finding</td>
<td>Directory Services Permission</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0031</td>
<td>Not a Finding</td>
<td>No Strong Name Verification</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0032</td>
<td>Not a Finding</td>
<td>First Match Code Groups</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0033</td>
<td>Not a Finding</td>
<td>File Code Groups, Net Code Groups</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0035</td>
<td>Not a Finding</td>
<td>Level Final Code Group Attribute</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0041</td>
<td>Not a Finding</td>
<td>Zone Membership Condition</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0045</td>
<td>Not a Finding</td>
<td>Administering CAS Policy</td>
<td>CAT I</td>
</tr>
<tr>
<td>APPNET0046</td>
<td>Not a Finding</td>
<td>Software Publishing Certificate</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0048</td>
<td>Not a Finding</td>
<td>Publisher Membership Condition</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0052</td>
<td>Not a Finding</td>
<td>Strong Name Membership Condition</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0054</td>
<td>Not a Finding</td>
<td>Administering CAS Policy for Group Names</td>
<td>CAT III</td>
</tr>
<tr>
<td>APPNET0055</td>
<td>Not a Finding</td>
<td>Administering CAS Policy and Policy Configuration File Backups</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0060</td>
<td>Not a Finding</td>
<td>Remoting Services Authentication and Encryption</td>
<td>CAT II</td>
</tr>
<tr>
<td>APPNET0061</td>
<td>Not a Finding</td>
<td>Unsupported .Net Framework Versions</td>
<td>CAT II</td>
</tr>
</tbody>
</table>

**Note:** APPNET0031, APPNET0033, APPNET0045, APPNET0046, APPNET0055, APPNET0060 and APPNET0061 must still be reviewed.

### 4.3.1 .NET Framework 1.0 Default Code Groups

```cmd
C:\ WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -lg
```

Microsoft (R) .NET Framework CasPol 1.0.3705.6018
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Code Groups:

1. All code: FullTrust

Level = Machine
UNCLASSIFIED

.NET Framework Security Checklist V1R3

22 April 2016

Developed by DISA for the DoD

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
       1.1.1. StrongName - 0024000004800009400000006020000002400000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AE8F87FB03766C834C99921EB23BE79AD9D5DC3C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
       1.1.2. StrongName - 0000000000000000000000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
       1.2.1. All code: Same site Web.
       1.2.2. All code: Same directory FileIO - Read, PathDiscovery
   1.3. Zone - Internet: Nothing
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
       1.5.1. All code: Same site Web.

Level = User

Code Groups:

1. All code: FullTrust
Success.

4.3.2  .NET Framework 1.1 Default Code Groups

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all –lg

Microsoft (R) .NET Framework CasPol 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Code Groups:

1. All code: FullTrust

Level = Machine

Code Groups:
1. All code: Nothing  
1.1. Zone - MyComputer: FullTrust  
1.1.1. StrongName - 
00240000048000094000000060200000024000005253413100040000100010007D1FA57C4AED9F0A32E84A0FAEFD0DE9E8FD6E9E8FC8F87FB03766C834C99921EB23BE79AD9D5DC  
C1D9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust  
1.1.2. StrongName - 000000000000000000000000000000000: FullTrust  
1.2. Zone - Intranet: LocalIntranet  
1.2.1. All code: Same site Web.  
1.2.2. All code: Same directory FileIO - Read, PathDiscovery  
1.3. Zone - Internet: Internet  
1.3.1. All code: Same site Web.  
1.4. Zone - Untrusted: Nothing  
1.5. Zone - Trusted: Internet  
1.5.1. All code: Same site Web.  

Level = User  

Code Groups:  

1. All code: FullTrust  
Success  

4.3.3 .NET framework 2.0, 3.0 & 3.5 Default Code Groups  
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -lg  
Microsoft (R) .NET Framework CasPol 2.0.50727.3053  
Copyright (c) Microsoft Corporation. All rights reserved.  

Security is ON  
Execution checking is ON  
Policy change prompt is ON  

Level = Enterprise  

Code Groups:  

1. All code: FullTrust  

Level = Machine
Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      002400000480000940000000620000002400000052534131000400000010010007D1FA57C4A
      ED9F0A32E84AA0FAEFDO0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
      1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63745D6F2DE5F17E5EAF0
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Nothing

Level = User

Code Groups:

1. All code: FullTrust

Success

4.3.4 .NET Framework 1.0 Default Permission Sets

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all –lp

Microsoft (R) .NET Framework CasPol 1.0.3705.6018
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet"
   version="1"
2. Skip Verification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="SkipVerification"
    Description="Grants right to bypass the verification">
<Permission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Execution"
    Description="Permits execution">
<Permission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Nothing"
    Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="LocalIntranet"
    Description="Default rights given to applications on the local intranet">
<Permission class="System.Security.Permissions.EnvironmentPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Read="USERNAME"/>
<Permission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<Permission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
6. Internet (Default rights given to internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Internet"
    Description="Default rights given to internet applications">
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
</PermissionSet>
Allowed="DomainIsolationByUser"
    UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Window="SafeTopLevelWindows"
    Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Level="SafePrinting"/>
</PermissionSet>

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in
permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib," Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
    ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
    ControlAppDomain, RemotingConfiguration, Infrastructure">
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
    System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
    System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
  version="1"
  Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  AllowBlankPassword="False"
  Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  AllowBlankPassword="False"
  Unrestricted="true"/>
</PermissionSet>

Level = Machine

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Unrestricted="true"
  Name="FullTrust"
  Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Name="SkipVerification"
  Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Name="Execution"
  Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Flags="Execution"/>
4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Nothing"
    Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="LocalIntranet"
    Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Allowed="AssemblyIsolationByUser"
    UserQuota="9223372036854775807"
    Expiry="9223372036854775807"
    Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIManagerPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"/>
6. Internet (Default rights given to internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Internet"
    Description="Default rights given to internet applications">
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
    <IPermission class="System.Security.Permissions.UILPermission, mscorlib, Version=1.0.3300.0,
        Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
    <IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in permissions">
    <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure"/>

<IPermission class="System.Security.Permissions.UIClassPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"/>
Level = User

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Unrestricted="true" Name="FullTrust"
    Description="Allows full access to all resources"/>

<PermissionSet class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<PermissionSet class="System.Diagnostics.PerformanceCounterPermission, System, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<PermissionSet class="System.DirectoryServices.DirectoryServicesPermission,
    System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<PermissionSet class="System.Messaging.MessageQueuePermission, System.Messaging,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<PermissionSet class="System.ServiceProcess.ServiceControllerPermission,
    System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
    PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<PermissionSet class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>

<PermissionSet class="System.Data.SqlClient.SqlClientPermission, System.Data,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>

</PermissionSet>
2. SkipVerification (Grants right to bypass the verification) =

```
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
Name="SkipVerification"
Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="SkipVerification"/>
</PermissionSet>
```

3. Execution (Permits execution) =

```
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
Name="Execution" Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1" Flags="Execution"/>
</PermissionSet>
```

4. Nothing (Denies all resources, including the right to execute) =

```
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>
```

5. LocalIntranet (Default rights given to applications on the local intranet) =

```
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
Name="LocalIntranet"
Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1" Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807" Permanent="True"/>
</PermissionSet>
```
6. Internet (Default rights given to internet applications) =

```xml
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Internet"
    Description="Default rights given to internet applications">
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
    <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
        Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
</PermissionSet>
```
7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```xml
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
Name="Everything"
Description="Allows unrestricted access to all resources covered by built-in permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure"/>
```
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
  PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
  System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
  PublicKeyToken=b03f5f7f11d50a3a"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.ServiceProcess.ServiceControllerPermission,
  System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
  PublicKeyToken=b03f5f7f11d50a3a"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  AllowBlankPassword="False"
  Unrestricted="true"/>
<Permission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>
</PermissionSet>

Success

4.3.5 .NET Framework 1.1 Default Permission Sets

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all –lp
Microsoft (R) .NET Framework CasPol 1.0.3705.6018
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet"
      version="1"
      Unrestricted="true" Name="FullTrust"
      Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
   <PermissionSet class="System.Security.NamedPermissionSet"
      version="1"
      Name="SkipVerification"
      Description="Grants right to bypass the verification">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
      Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
      version="1"
      Flags="SkipVerification"/>
   </PermissionSet>

3. Execution (Permits execution) =
   <PermissionSet class="System.Security.NamedPermissionSet"
      version="1"
      Name="Execution"
      Description="Permits execution">
<PermissionSet class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Name="Nothing"
  Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Name="LocalIntranet"
  Description="Default rights given to applications on the local intranet">
  <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Read="USERNAME"/>
  <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
  <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Allowed="AssemblyIsolationByUser"
    UserQuota="9223372036854775807"
    Expiry="9223372036854775807"
    Permanent="True"/>
  <IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="ReflectionEmit"/>
  <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, Execution"/>
  <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    Unrestricted="true"/>
  <IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"/>
6. Internet (Default rights given to internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Name="Internet"
  Description="Default rights given to internet applications">
  <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Access="Open"/>
  <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Allowed="DomainIsolationByUser"
    UserQuota="10240"/>
  <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Execution"/>
  <IPermission class="System.Security.Permissions.UIManagerPermission, mscorlib, Version=1.0.3300.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Window="SafeTopLevelWindows"
    Clipboard="OwnClipboard"/>
  <IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Unrestricted="true"/>
```

Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
<PermissionSet Name="FullTrust" Level="Machine">
  <IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <IPermission class="System.Diagnostics.PerformanceCounterPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <IPermission class="System.DirectoryServices.DirectoryServicesPermission, System.DirectoryServices, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true"/>
  <IPermission class="System.Messaging.MessageQueuePermission, System.Messaging, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true"/>
  <IPermission class="System.ServiceProcess.ServiceControllerPermission, System.ServiceProcess, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true"/>
  <IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" AllowBlankPassword="False" Unrestricted="true"/>
  <IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" AllowBlankPassword="False" Unrestricted="true"/>
</PermissionSet>

**Level = Machine**

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
    Unrestricted="true" Name="FullTrust"
    Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
    Name="SkipVerification"
    Description="Grants right to bypass the verification">
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
    Name="Execution"
    Description="Permits execution">
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
    Name="Nothing"
    Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
    Name="LocalIntranet"
    Description="Default rights given to applications on the local intranet">
    <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
6. Internet (Default rights given to internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
  version="1"
  Name="Internet"
  Description="Default rights given to internet applications">
  <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Access="Open"/>
  <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
  Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Allowed="AssemblyIsolationByUser"
    UserQuota="9223372036854775807"
    Expiry="9223372036854775807"
    Permanent="True"/>
  <IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
  PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
  <IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    <Machine name=".
      access="Instrument"/>
  </PermissionSet>
7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
    Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure"/>

<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Diagnostics.PerformanceCounterPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.DirectoryServices.DirectoryServicesPermission, System.DirectoryServices, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>
</PermissionSet>

**Level = User**

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet"
       version="1"
       Unrestricted="true" Name="FullTrust"
       Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
   <PermissionSet class="System.Security.NamedPermissionSet"
       version="1"
       Name="SkipVerification"
       Description="Grants right to bypass the verification">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
       Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
       version="1"
       Flags="SkipVerification"/>
   </PermissionSet>

3. Execution (Permits execution) =
   <PermissionSet class="System.Security.NamedPermissionSet"
       version="1"
       Name="Execution"
       Description="Permits execution">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
       Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
       version="1"
       Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Nothing"
    Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="LocalIntranet"
    Description="Default rights given to applications on the local intranet">
    <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib," Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib," Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib," Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="AssemblyIsolationByUser"
        UserQuota="9223372036854775807"
        Expiry="9223372036854775807"
        Permanent="True"/>
    <IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib," Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="ReflectionEmit"/>
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib," Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, Execution"/>
    <IPermission class="System.Security.Permissions.UI Permission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
6. Internet (Default rights given to internet applications) =
   <PermissionSet class="System.Security.NamedPermissionSet"
      version="1"
      Name="Internet"
      Description="Default rights given to internet applications">
   <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
      Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
      version="1"
      Access="Open"/>
   <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
      Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
      version="1"
      Allowed="DomainIsolationByUser"
      UserQuota="10240"/>
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
      Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
      version="1"
      Flags="Execution"/>
   <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
      Culture=neutral, PublicKeyToken=b77a5c561934e089"
      version="1"
      Window="SafeTopLevelWindows"
      Clipboard="OwnClipboard"/>
   <IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
      Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
      version="1"
      Level="SafePrinting"/>
   </PermissionSet>

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
   <PermissionSet class="System.Security.NamedPermissionSet"
      version="1"
      Name="Everything"
      Description="Allows unrestricted access to all resources covered by built-in permissions">
   <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
      Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
      version="1"
      Level="SafePrinting"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure"/>

<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
  version="1"
  Unrestricted="true"/>

<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    AllowBlankPassword="False"
    Unrestricted="true"/>
</PermissionSet>

Success

4.3.6 .NET Framework 2.0, 3.0 & 3.5 Default Permission Sets
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all –lp

Microsoft (R) .NET Framework CasPol 2.0.50727.3053
Copyright (c) Microsoft Corporation. All rights reserved.
UNCLASSIFIED

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Unrestricted="true"
    Name="FullTrust"
    Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
   <PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="SkipVerification"
    Description="Grants right to bypass the verification">
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="SkipVerification"/>
   </PermissionSet>

3. Execution (Permits execution) =
   <PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Execution"
    Description="Permits execution">
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Execution"/>
   </PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
   <PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Nothing"
    Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
   <PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
Name="LocalIntranet"
   Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   Allowed="AssemblyIsolationByUser"
   UserQuota="9223372036854775807"
   Expiry="9223372036854775807"
   Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   Flags="Assertion, Execution, BindingRedirects"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
   PublicKeyToken=b77a5c561934e089"
   version="1"
   Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
   version="1"
   Level="DefaultPrinting"/>
</PermissionSet>

6. Internet (Default rights given to Internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
   version="1"
   Name="Internet"
   Description="Default rights given to Internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in permissions">
    <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
</PermissionSet>
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIManagerPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.KeyContainerPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
    version="1" Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
Level = Machine

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet"
   version="1"
   Unrestricted="true"
   Name="FullTrust"
   Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
   <PermissionSet class="System.Security.NamedPermissionSet"
   version="1"
   Name="SkipVerification"
   Description="Grants right to bypass the verification">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   Flags="SkipVerification"/>
   </PermissionSet>

3. Execution (Permits execution) =
   <PermissionSet class="System.Security.NamedPermissionSet"
   version="1"
   Name="Execution"
   Description="Permits execution">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
   version="1"
   
   "UNCLASSIFIED"
Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="LocalIntranet"
Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807"
Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Assertion, Execution, BindingRedirects"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
6. Internet (Default rights given to Internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Internet"
    Description="Default rights given to Internet applications">
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="ApplicationIsolationByUser"
        UserQuota="512000"/>
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
    <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
        Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
    <IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in permissions">
    <IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
</PermissionSet>
<IPermission class="System.Security.Primitives.FileIOPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Primitives.IsolatedStorageFilePermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Primitives.ReflectionPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Primitives.RegistryPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Primitives.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure"/>

<IPermission class="System.Security.Primitives.UIPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Security.Primitives.KeyContainerPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.SocketPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>

<IPermission class="System.Net.WebPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Unrestricted="true">
  <Permission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <Permission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <Permission class="System.Diagnostics.PerformanceCounterPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <Permission class="System.Data.OleDb.OleDbPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <Permission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <Permission class="System.Security.Permissions.DataProtectionPermission, System.Security, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true"/>
</PermissionSet>

Level = User

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
   <PermissionSet class="System.Security.NamedPermissionSet" version="1" Unrestricted="true">
     <Permission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Diagnostics.PerformanceCounterPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Data.OleDb.OleDbPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Security.Permissions.DataProtectionPermission, System.Security, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true"/>
   </PermissionSet>

2. SkipVerification (Grants right to bypass the verification) =
   <PermissionSet class="System.Security.NamedPermissionSet" version="1" Unrestricted="true">
     <Permission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Diagnostics.PerformanceCounterPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Data.OleDb.OleDbPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
     <Permission class="System.Security.Permissions.DataProtectionPermission, System.Security, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true"/>
   </PermissionSet>
Flags="SkipVerification" />

3. Execution (Permits execution) = 
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Execution"
    Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Nothing"
    Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="LocalIntranet"
    Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Allowed="AssemblyIsolationByUser"
    UserQuota="9223372036854775807"
    Expiry="9223372036854775807"
    Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Reflection Emit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
6. Internet (Default rights given to Internet applications) =

```xml
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Internet"
    Description="Default rights given to Internet applications">
    <IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
    <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="ApplicationIsolationByUser"
        UserQuota="512000"/>
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
    <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
    <IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =
<PermissionSet class="System.Security.NamedPermissionSet"
    version="1"
    Name="Everything"
    Description="Allows unrestricted access to all resources covered by built-in permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure, BindingRedirects"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Security.Permissions.KeyContainerPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"
    Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    version="1"/>
Success

4.4 IAVM Compliance

IAVM alerts, bulletins, and advisories were instituted to provide positive control of vulnerability notification and corresponding corrective action within DOD. All DOD program managers and system administrators, and/or other personnel responsible for system networks shall comply with the IAVM process. Security patches that address .NET vulnerabilities are
reviewed during an operating system security review and are not included in this checklist.

4.5 Version-specific Vulnerabilities

Vulnerabilities that apply only to a specific version of .NET are so noted. Versions to which the vulnerability does not apply should have that finding marked as N/A.

4.6 .NET Configuration File Location

<table>
<thead>
<tr>
<th>Level</th>
<th>Version</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise</td>
<td>1.0</td>
<td>%SystemRoot%\Microsoft.NET\Framework\Config\v1.0.3705\EnterpriseSec.config</td>
</tr>
<tr>
<td>Enterprise</td>
<td>1.1</td>
<td>%SystemRoot%\Microsoft.NET\Framework\Config\v1.1.4322\EnterpriseSec.config</td>
</tr>
<tr>
<td>Enterprise</td>
<td>2.0, 3.0 &amp; 3.5</td>
<td>%SystemRoot%\Microsoft.NET\Framework\Config\v2.0.50727\EnterpriseSec.config</td>
</tr>
<tr>
<td>Machine</td>
<td>1.0</td>
<td>%SystemRoot%\Microsoft.NET\Framework\Config\v1.0.3705\Security.config</td>
</tr>
<tr>
<td>Machine</td>
<td>1.1</td>
<td>%SystemRoot%\Microsoft.NET\Framework\Config\v1.1.4322\Security.config</td>
</tr>
<tr>
<td>Machine</td>
<td>2.0, 3.0 &amp; 3.5</td>
<td>%SystemRoot%\Microsoft.NET\Framework\Config\v2.0.50727\Security.config</td>
</tr>
<tr>
<td>User</td>
<td>1.0</td>
<td>User Profile\Application Data\Microsoft\CLR Security Config\v1.0.3705</td>
</tr>
<tr>
<td>User</td>
<td>1.1</td>
<td>User Profile\Application Data\Microsoft\CLR Security Config\v1.1.4322</td>
</tr>
<tr>
<td>User</td>
<td>2.0, 3.0 &amp; 3.5</td>
<td>User Profile\Application Data\Microsoft\CLR Security Config\v2.0.50727.42</td>
</tr>
</tbody>
</table>

**Note:** Microsoft .Net versions 3.0 and 3.5 utilize the .Net 2.0 CLR and the same security configuration files.

**Note:** If configuration files are missing or corrupt the default configuration will be applied.

4.7 Reviewing Permissions with Code Access Security Policy Tool

The .NET Framework Code Access Security (CAS) Policy Tool caspol.exe may be used to review the .NET Framework code access security configuration. Each version of the .Net Framework comes with its own version of caspol.exe. Each version of caspol.exe can only be used to administer the .Net Framework version for which it was built. Use the version of caspol.exe found in the same directory structure as the .NET Framework version that is being reviewed. Security configurations for .Net 3.0 and 3.5 use the same underlying CLR and share configuration information as .Net 2.0.
Note: Any findings found in .Net 2.0 should be marked in .Net 3.0 and .Net 3.5 Frameworks if they are installed.

Following are caspol.exe command line syntaxes for displaying the .NET code groups, permission sets, and trust assemblies. Issue all commands from the Windows command line accessed via the Windows Start>>Run>> open: cmd. Output may be directed to a text file with the use of the > redirection pipe. The resulting file may then be reviewed using the Windows Notepad or other text editor.

To list code groups for all levels, type the following command.

For .Net Framework Versions 1.0

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -lg

For .Net Framework Versions 1.1

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all -lg

For .Net Framework Versions 2.0, 3.0 & 3.5

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -lg

To list code group names and descriptions for all levels, type the following command.

For .Net Framework Versions 1.0

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -ld

For .Net Framework Versions 1.1

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all -ld

For .Net Framework Versions 2.0, 3.0 & 3.5

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -ld

To list permission sets for all levels, type the following command.

For .Net Framework Versions 1.0

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -lp

For .Net Framework Versions 1.1
C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all -lp

For .Net Framework Versions 2.0, 3.0 & 3.5

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -lp

Caspol.exe can typically be found in the %systemroot%\Microsoft.NET\Framework\<version>\ directory.

Microsoft .Net version 2.0, 3.0 & 3.5 use the same underlying Common Language Runtime (CLR) is and share the same configuration files.

4.8 Reviewing Software Publishing State Values

Use regedit to determine the value of the registry keys for all users of the machine. All values for software publishing should be reviewed.

Example Locations:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

HKEY_USERS\S-1-5-18\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

HKEY_USERS\S-1-5-19\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

HKEY_USERS\S-1-5-20\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State
Figure 4-1: Review Software Publishing State Values

Convert the State DWORD value to binary with the hexadecimal calculator.

For Example:

If the registry setting is 23c00 Hex it converts to 00100011110000000000 Binary with 2 leading zeros for 19 bits total.

Use Table 4-4 to determine the individual values for software publishing.
### Table 4-4: Software Publishing State Value Table

<table>
<thead>
<tr>
<th>Hex</th>
<th>2</th>
<th>3</th>
<th>C</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Bit 19 = 1 Only trust items found in the Trust DB</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 18 = 1 Check the revocation list on Time Stamp Signer</td>
<td>FALSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 17 = 1 Invalidate version 1 signed objects =</td>
<td>FALSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 14 = 1 Java offline revocation server OK (Commercial) =</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 13 = 1 Java offline revocation server OK (Individual) =</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 12 = 1 Offline revocation server OK (Commercial) =</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 11 = 1 Offline revocation server OK (Individual) =</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bit 10 = 1 Check the revocation list =</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bits 6 &amp; 8 = 1 Trust the Test Root =</td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Bits 6 &amp; 8 = 0 Trust the Test Root =</td>
<td>FALSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Only trust items found in the Trust DB = FALSE
Check the revocation list on Time Stamp Signer = FALSE
Invalidate version 1 signed objects = FALSE
Java offline revocation server OK (Commercial) = TRUE
Java offline revocation server OK (Individual) = TRUE
Offline revocation server OK (Commercial) = TRUE
Offline revocation server OK (Individual) = TRUE
Check the revocation list = TRUE
5. .NET SECURITY FRAMEWORK CHECKS AND PROCEDURES

The following instructions should be used for checks APPNET0001 through APPNET0030. Use the permission name specified in the individual instruction to determine the specific permission to review. Any instructions included under a check are specific to that check and should be followed.

Performing a SRR on the .NET Framework involves identifying which permission sets have potentially dangerous permissions. After determining which permission sets are of interest the code groups to which those permission sets are assigned must be identified. Once the code groups are identified the membership conditions for those code groups must be documented. When determining whether or not a given vulnerability exists the reviewer must evaluate the membership conditions of code groups that grant the permission and determine if the permission is restricted to the appropriate assemblies. When entering the finding details the reviewer should include the name of the permission set(s) and the code group(s) that grant the permission set(s).

5.1 APPNET0001: File IO Permission

Description: The File IO permission allows an application to access system files directly.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.FileIOPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
002400000480000094000000060200000240000052534131000400000100010007D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6E834C9921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF80957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 000000000000000040000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If a Permission Set with the File IO permission of Grant Assemblies unrestricted access to the file system (unrestricted="true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If a Permission Set granting limited File IO permissions is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>File IO Permission</td>
<td></td>
</tr>
</tbody>
</table>

5.2 APPNET0003: Isolated Storage Permission

Description: The Isolated Storage permission is used to allow applications to store temporary data to a local user data store.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.IsolatedStorageFilePermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName - 00240000048000000940000006020000024000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFDD0DE9E8FD6AE8F87FB03766C834C99921EB23BE79AD9D5DC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
1. If the Isolated Storage permission of Grant assemblies unrestricted access to file-based storage (unrestricted="true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the Isolated Storage permission Administer Isolated Storage by User is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
3. If the Isolated Storage permissions Assembly Isolation by User or Assembly Isolation by Roaming User is assigned to a Non-default Code Group whose membership criteria has not been evaluated and approved by the ISSO then this is a finding.
### 5.3 APPNET0004: User Interface Permission (Windowing)

**Description:** The User Interface Permission for windowing controls access to user interface windows.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission `System.Security.Permissions.UIPermission` for Windows.

**Example:**
```
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Window="SafeSubWindows" Clipboard="OwnClipboard" />
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

**Example:**

**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName - 0024000004800000940000006020000002400005253413100400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB9285D622CAA652C1DFAD63D745D6F2DE5F17FE5EAF0FC49632D61C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000004000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:

1. If the User Interface permission of *Grant assemblies unrestricted access to user interface elements (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

2. If the User Interface permission *All Windows Events* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

3. If the User Interface permissions *Safe Top Level Windows* is assigned to a Non-default Code Group whose membership criteria has not been evaluated and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>User Interface Permission (Windowing)</td>
<td></td>
</tr>
</tbody>
</table>

5.4 **APPNET0005: User Interface Permission (Clipboard)**

**Description:** The User Interface Permission for clipboard controls application access to clipboards used by the user or other applications.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.UIPermission for Clipboard.

Example:

7. Dev =

```xml
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
              Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Window="SafeSubWindows" Clipboard="OwnClipboard"/>
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.
Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
       1.1.1. StrongName - 0024000004800000940000000602000000240000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AE8C8F87FB03766C834C99921EB23BE79AD9D5DC1DD9AD236123102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821CA05EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
       1.1.2. StrongName - 0000000000000000040000000000000000400000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
       1.2.1. All code: Same site Web
   1.3. Zone - Internet: Internet
       1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
       1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
1. If any User Interface permission other than No Clipboard is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

| Category: | CAT II | Level: Gold MCL: 1-CSP;2-CSP;3-
| SDID: | User Interface Permission (Clipboard) | |

5.5 APPNET0006: Reflection Permission

Description: The Reflection permission controls an application's discovery of other system resources and applications.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.ReflectionPermission.
Example:
7. Dev =

```xml
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
    Unrestricted="true" />
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
          002400000480000940000000602000000240000525341310004000001001007D1FA57C4A
          ED9F0A32E84AA0FAEFDODE9E8FD6AEC8F87FB03766C834C9921EB23BE79AD9D5DC
          C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
          1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63745D6F2DE5F17E5EAF0
          FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

**Validate:**
1. If the Reflection permission of Grant assemblies unrestricted permission to discover information about other assemblies (unrestricted=true) is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the Reflection permission Member is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
3. If the Reflection permissions Type is assigned to a Non-default Code Group whose membership criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Reflection Permission</td>
<td></td>
</tr>
</tbody>
</table>

5.6 APPNET0007: Printing Permission

**Description:** The Printing permission controls application access to system printing resources.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Drawing.Printing.PrintingPermission.

Example:

7. Dev =  
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Drawing.Printing.PrintingPermission", System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1"  
  Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

**Code Groups:**

1. All code: Nothing  
   1.1. Zone - MyComputer: FullTrust  
      1.1.1. StrongName -  
      002400000048000009400000006020000024000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AE8F87FB03766C834C99921EB23BE79AD9D5DC  
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82  
      1C0A5EFE88F1645C4C0C93C1AB99285D622C9A652C1DFAD63D745D6F2DE5F17E5EAF0  
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust  
      1.1.2. StrongName - 000000000000004000000000000000: FullTrust  
1.2. Zone - Intranet: LocalIntranet  
   1.2.1. All code: Same site Web  
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If the Printing permission of Grant assemblies unrestricted access to printers (unrestricted= "true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the Printing permission All Printing is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

| Category: | CAT II | Level: Gold MCL: 1-CSP;2-CSP;3-
| SDID : | Printing Permission |

5.7 APPNET0008: DNS Permission

Description: The DNS permission controls application access to DNS resources available to the host system.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Net.DnsPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
   <IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:
1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      0024000004800000940000006020000002400000525341310004000001000100007D1FA57C4A
      ED9F0A32E84AA0FEF0DE9E8F87F03766C834C9921EB23BE79AD9D5DC
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
      1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000040000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
   1. If the DNS permission of Grant assemblies unrestricted access to DNS
      (unrestricted="true") is assigned to a Non-default Code Group that does not use a
      Strong Name, Publisher, or Hash as the membership condition and whose assignment
      criteria has not been reviewed and approved by the ISSO then this is a finding.

| Category: | CAT II | Level: Gold MCL: 1-CSP;2-CSP;3- |
| SDID: DNS Permission |

5.8 APPNET0009: Socket Access Permission

Description: The Socket Access permission controls application access to network ports defined
on the host system.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
   <IPermission class="System.Net.SocketPermission", System, Version=2.0.0.0,
      Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      00240000048000009400000060200000024000052534131000400000100010007D1FA57C4A
      ED9FA032E84AA0FAEF0DDE9E8FD6AE810E87FB03766C834C99921EB23BE79AD9D5DC
      C1DD9AD236132102900B723CF80957FC4E177108FC607774F29E8320E92EA05ECE4E82
      1C0A5FE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 0000000000000000040000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:

1. If the **Socket Access** permission of **Grant assemblies unrestricted access to sockets**
   (unrestricted="true") is assigned to a Non-default Code Group that does not use a
   Strong Name, Publisher, or Hash as the membership condition and whose assignment
   criteria has not been reviewed and approved by the ISSO then this is a finding.
2. Ask the System Administrator if any **Socket Access** permissions are granted to Non-
   default Code groups that do not provide networking services. If these permissions exist
   then this is a finding.
3. Ask the System Administrator if any **Socket Access** permissions are granted to Non-
   default Code groups to hosts outside the enclave. If these permissions exist then this is a
   finding.

Category: **CAT II**

SDID: **Socket Access Permission**


IA Control: **DCSL-1**
5.9 APPNET0010: Web Access Permission

**Description:** The Web Access permission controls application access to HTTP requests to designated URLs or the configuration of HTTP settings.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Net.WebPermission.

Example:

```
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Net.WebPermission", System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a561934e089" version="1" Unrestricted="true" />
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
         00240000048000009400000006020000024000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFDODE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8FB645C40C93C1AB99285D622C662C1DFAD63D7456F2DF5E517E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
         1.1.2. StrongName - 0000000000000000040000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

**Validate:**

1. If the Web Access permission of Grant assemblies unrestricted access to Web Sites
(unrestricted="true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

2. If specific URL(s) (Web Access permissions) are assigned to a Non-default Code Group whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID: APPNET10</td>
<td>Web Access Permission</td>
<td></td>
</tr>
</tbody>
</table>

5.10 APPNET0011: Message Queue Permission

**Description:** The Message Queue permission controls application access to communications across the network.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Messaging.MessageQueuePermission.

Example:

7. Dev =

```xml
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Messaging.MessageQueuePermission, System.Messaging, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1"
  Unrestricted="true" />
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      00240000004800000940000006020000000240000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD66AEC8F87FB03766C834C99921EB23BE79AD95DC
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05CE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If the Message Queue permission of Grant assemblies unrestricted access to all message queues (unrestricted="true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the Message Queue permission Administer is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
3. If the Message Queue permission Browse is assigned to a Non-default Code Group whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID : Message Queue Permission</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.11 APPNET0012: Service Controller Permission

Description: The Service Controller permission controls application access to the control of Windows services.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.ServiceProcess.ServiceControllerPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
   <IPermission class="System.ServiceProcess.ServiceControllerPermission, System.ServiceProcess, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true" />
</PermissionSet>
Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName - 0024000004800009400000006020000024000052534131000400000100010007D1FA57C4A ED9F0A32E84AA0FAEFDD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82 1CA5EFE8FI645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0 FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000400000000000000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

**Validate:**

1. If the any *Service Controller* permissions are assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

**Category:** CAT II  
**Level:** Gold MCL: 1-CSP; 2-CSP; 3-

**SDID :** Service Controller Permission

**Reference:** .NET Framework Security Guide

**IA Control:** DCSL-1

**5.12 APPNET0013: Database Permission**

**Description:** The Database permissions control application access to databases defined on the host system.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5
Use caspol to review the non-default permission sets with the permission System.Data.SqlClient.SqlClientPermission or System.Data.OleDb.OleDbPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Data.SqlClient.SqlClientPermission", System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
  <IPermission class="System.Data.OleDb.OleDbPermission", System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true"/>
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName - 0024000004800000940000006020000024000052534131000400000100010007D1FA57C4AED9F0A32E84A0FAEFD0DE9E8FD6AE8F87FB03766C834C99921EB23BE79AD9D5DC C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0 FC4963D261C8A12436518206DC903344D5AD293: FullTrust
      1.1.2. StrongName - 000000000000000000000040000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
1. If the SQLClientPermission or OLEDBPermission permission (Grant assemblies unrestricted access to all providers (unrestricted="true") is assigned to a Non- default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
Category: CAT II  
Level: Gold MCL: 1-CSP; 2-CSP; 3-

SDID: Database Permission  
IA Control: DCSL-1

Checks APPNET014 – APPNET025 refer to the Security permission contained within a permission set assigned to a non-default code group. Review all permission sets that include the security permission.

5.13 APPNET0014: Security Permission (Extend Infrastructure)

Description: The Security permission Extend Infrastructure controls application access to message processing.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:

7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">  
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,  
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"  
Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,  
ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,  
ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />  
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

Level = Machine

Code Groups:

1. All code: Nothing  
   1.1. Zone - MyComputer: FullTrust  
      1.1.1. StrongName -
      00240000048000094000000602000000240000052534131000400000100010007D1FA57C4A  
      ED9F0A32E84AA0FAEF0D0DE9E8FD6AEC8F87FB03766C834C99921EB23BE9AD9D5DC  
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82  
      1C0A5EF8E8F1645C40C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0  
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 000000000000000004000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If the Security permission of Extend Infrastructure is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Security Permission (Extend Infrastructure)</td>
</tr>
<tr>
<td>Level: Gold MCL:</td>
<td>1-CSP;2-CSP;3-</td>
</tr>
</tbody>
</table>

5.14 APPNET0015: Security Permission (Enable Remoting Configuration)

Description: The Security permission Enable Remoting Configuration defines the communication channels available to an application.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =

<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
         0024000004800000940000000602000000024000005253413100040000010000007D1FA57C4A
         ED9F0A32E84A0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
         C1DD9AD23613210290B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
         1C0A5EFE8F1645C4C0C93C1AB99285D622C6AA652C1DFAD63D745D6F2DE5F17E5EAF0
         FC4963D261CA12436518206DC09334AD5AD293: FullTrust
         1.1.2. StrongName - 000000000000000000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
   1. If the Security permission of Enable remoting configuration
      (Flags="RemotingConfiguration") is assigned to a Non-default Code Group that does
      not use a Strong Name, Publisher, or Hash as the membership condition and whose
      assignment criteria has not been reviewed and approved by the ISSO then this is a
      finding.

| Category:  | CAT II | Level: Gold MCL: 1-CSP;2-CSP;3- |
| SDID:      | Security Permission (Enable Remoting configuration) |

5.15 APPNET0016: Security Permission (Enable Serialization Formatter)

Description: The Security permission Enable Serialization Formatter controls access to
serialized data. Serialized data is data formatted into a series of bits for storing or transmitting.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission
Example:
7. Dev =

```xml
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
        Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
       1.1.1. StrongName -
           00240000048000009400000062000000240005253413100040000010001007D1FA57C4A
           ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
           C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
           1C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EA0F
           FC4963D261C8A12436518206DC093344D5AD293: FullTrust
       1.1.2. StrongName - 00000000000000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
       1.2.1. All code: Same site Web
       1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
       1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
       1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

**Validate:**
1. If the Security permission of *Enable Serialization Formatter (Flags="SerializationFormatter")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
Category: CAT II  
Level: Gold MCL: 1-CSP; 2-CSP; 3-CSP; 4-CSP; 5-CSP

SDID: Security Permission (Enable Serialization Formatter)  
IA Control: DCSL-1

**5.16 APPNET0017: Security Permission (Enable Thread Control)**

**Description:** The Security permission Enable Thread Control is used to control application access to abort, suspend, or resume its threads.

**Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5**

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:

7. Dev =

```
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, **ControlThread**, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>
```

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName - 002400000480000094000000602000000240000052534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC1D9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A2436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000000D400000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. Application Directory: Dev

Validate:
1. If the Security permission of Enable thread control (Flags="ControlThread") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-CSP;4-CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Security Permission (Enable Thread Control)</td>
<td></td>
</tr>
</tbody>
</table>

5.17 APPNET0018: Security Permission (Allow Principal Control)

Description: The Security permission Allow Principal control controls application access to Windows user information.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
   <IPermission class="System.Security.Permissions.SecurityPermission", mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
   Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,
   ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,
   ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:
1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
   1.1.1. StrongName -
   00240000048000009400000006020000024000052534131004000010010007D1FA57C4A
   ED9F0A32E84A0FAEDFD0DE9E8FD6AE8C884C99221EB23BE79AD9D5DC
   C1DD9AD236132102900B723CF980957FC4E17108FC67774F29E8320E92EA05EC4E82
   1CA5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD637D456F2DE5F17E5EAF0
   FC4963D261C8A12436518206DC093344D5AD93: FullTrust
   1.1.2. StrongName - 00000000000000000000000000000000: FullTrust

1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web

1.4. Zone - Untrusted: Nothing

1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web

1.6. ApplicationDirectory: Dev

Validate:

1. If the Security permission of Allow principal control (Flags="ControlPrincipal") is
   assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or
   Hash as the membership condition and whose assignment criteria has not been reviewed
   and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold</th>
<th>MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Security Permission (Allow Principal Control)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.18 APPNET0019: Security Permission (Enable Assembly Execution)

Description: The Security permission Enable Assembly Execution allows applications to
execute.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission

Example:

7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
   <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
   Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1">
   Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,
   ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,
ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      0024000004800000940000006020000002400000525341310004000010001007D1FA57C4A
ED9F0A32E84A0FAEFDD0DE9E8FD6AEC08F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAE0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
1. If the Security permission of Enable assembly execution (Flags="Execution") is in a permission set that is assigned to a Non-default Code Group with a Zone membership condition then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Security Permission (Enable Assembly Execution)</td>
<td></td>
</tr>
</tbody>
</table>

5.19 APPNET0020: Security Permission (Skip Verification)

Description: The Security permission Skip Verification controls the execution of code that is verified as being type safe.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5
Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="System.Security.Permissions.SecurityPermission",mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,
ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,
ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
0024000004800009400000006020000052534131000400000100010007D1FA57C4A
ED9F0A32E84AA0FAAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000400000000000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If the Security permission of Skip verification (Flags="SkipVerification") is assigned to any non-default Code Group then this is a finding.
5.20 APPNET0021: Security Permission (Allow Calls to Unmanaged Assemblies)

**Description:** The Security permission Allow Calls to Unmanaged Assemblies controls application access to applications not managed by the .Net Framework.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">  
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,  
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"  
Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,  
ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,  
ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />  
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
**Level = Machine**

**Code Groups:**

1. All code: Nothing  
   1.1. Zone - MyComputer: FullTrust  
      1.1.1. StrongName =
      00240000048000009400000006020000024000052534131000400000100010007D1FA57C4A
      ED9F0A32E84A0F0AED0DE9E8FD687FB003766834C99921EB23BE79AD9D5DC
      C1DD9AD236132102900B723CF980957FC4E177108FC607754F29E8320E92EA05ECE4E82
      1C0A5EFE8F1645C40C93C1AB9285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust  
      1.1.2. StrongName = 00000000000000000000004000000000000000000: FullTrust  
   1.2. Zone - Intranet: LocalIntranet  
      1.2.1. All code: Same site Web  
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'  
   1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If the Security permission of Allow calls to unmanaged assemblies
   (Flags="UnmanagedCode") is assigned to a non-default Code Group then this is a
   finding.

Category: CAT II  Level: Gold MCL: 1-CSP;2-CSP;3-
SDID: Security Permission (Allow Calls to Unmanaged Assemblies)

5.21 APPNET0022: Security Permission (Allow Policy Control)

Description: The Security permission Allow Policy Control controls application access to it’s
the current security policy configuration.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission

Example:
7. Dev =
  <PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
    <IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,
ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,
ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
  </PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to
determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
00240000048000009400000006200000024000005253413100400000100010007D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AE8C8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD23612102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1CA5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 00000000000000000400000000000000000000: FullTrust

1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web

1.4. Zone - Untrusted: Nothing

1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web

1.6. ApplicationDirectory: Dev

Validate:

1. If the Security permission of Allow Policy Control (Flags="ControlPolicy") is assigned
to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the
membership condition and whose assignment criteria has not been reviewed and
approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP; 2-CSP; 3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Security Permission (Allow Policy Control)</td>
<td>IA Control: DCSL-1</td>
</tr>
</tbody>
</table>

5.22 APPNET0023: Security Permission (Allow Domain Policy Control)

Description: The Security permission Allow Domain Policy controls defines application access
to its own application domain security policy.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission

Example:

7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  mscorlib,
  Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
  Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread,
  ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,
  ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
       1.1.1. StrongName -
           002400000480000009400000060200000024000005253413100040000010001007D1FA57C4A
           ED9F0A32E84AA0FAEF0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
           C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
           1C0A5EFE8F1645C40C93C1AB99285D622CAA652C1DFAD63745D6F2DE5F17E5EAF0
           FC4963D261C8A12436518206DC09334D5AD293: FullTrust
           1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
       1.2.1. All code: Same site Web
       1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
       1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
       1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

**Validate:**

1. If the Security permission of *Allow domain policy control* 
   *(Flags = "ControlDomainPolicy")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Security Permission (Allow Domain Policy Control)</td>
<td></td>
</tr>
</tbody>
</table>

**5.23 APPNET0024: Security Permission (Allow Evidence Control)**

**Description:** The Security permission Allow Evidence Control is used to control an application's access to supply or modify evidence used to determine access to system resources.
Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Security.Permissions.SecurityPermission", mscorlib, 
  Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" 
  Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, 
  ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, 
  ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
       1.1.1. StrongName - 0024000004800094000000602000002400005253413100040000010001007D1FA57C4AED9F0A32E84A0FAFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5A5D293: FullTrust
       1.1.2. StrongName - 0000000000000000000040000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
       1.2.1. All code: Same site Web
   1.3. Zone - Internet: Internet
       1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
       1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
1. If the Security permission of Allow evidence control (Flags="ControlEvidence") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed
and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Security Permission (Allow Evidence Control)</td>
<td></td>
</tr>
</tbody>
</table>

5.24 APPNET0025: Security Permission (Assert any Permission that Has Been Granted)

Description: The Security permission Assert any Permission that Has Been Granted controls application access to permissions assigned to any code in the assembly that called it.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev"><IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      00240000004800009400000006020000024000052534131000400000100010007D1FA57C4AED9F0A32E84A0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC1C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA0E5ECE4E821C0A5EE8F1645C4C0C93C1AB99285622C4A652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D61C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000004000000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
  1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
  1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate
1. If the Security permission of Assert (Flags="Assertion") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Security Permission (Assert any Permission that Has Been Granted)</td>
<td></td>
</tr>
</tbody>
</table>

5.25 APPNET0026: Performance Counter Permission

Description: The Performance Counter permission controls application access to system performance monitoring resources.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Diagnostics.PerformanceCounterPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="System.Diagnostics.PerformanceCounterPermission", System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1">
    <Machine name="server1">
      <Category name="admin" access="Administer" />
    </Machine>
  </IPermission>
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine
Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
         002400000048000009400000006200000024000052534131000400000100010007D1FA57C4A
         ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
         C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
         1C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0
         FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 0000000000000000040000000000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

Validate:
   1. If the Performance Counter is assigned to a Non-default Code Group that does not use a
      Strong Name, Publisher, or Hash as the membership condition and whose assignment
      criteria has not been reviewed and approved by the ISSO then this is a finding.

   | Category: CAT II | Level: Gold MCL: 1-CSP;2-CSP;3-
   | SDID : Performance Counter Permission | IA Control: ECLP-1 |

5.26 APPNET0027: Environment Variables Permission

Description: The Environment Variables permission controls application access to system
environment variables and to other system resource names.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission

Example:
7. Dev =
   <PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
      <IPermission class="System.Security.Permissions.EnvironmentPermission", mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName - 002400000480000094000000602000000240000052534131000040000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
      1.1.2. Zone - Intranet: LocalIntranet
      1.1.2.1. All code: Same site Web
      1.1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. ApplicationDirectory: Dev

**Validate:**

1. If the Environment Variables permission of Grant assemblies access to all environment variables (unrestricted="true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Environment Variables Permission</td>
<td></td>
</tr>
</tbody>
</table>

**5.27 APPNET0028: Event Log Permission**

**Description:** The Event Log permission controls application access to event log resources defined on the system.
Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Diagnostics.EventLogPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
    <IPermission class="System.Diagnostics.EventLogPermission", System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5e561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
   1.1.1. StrongName -
   0024000004800009400000062000000024000052534131000400000100010007D1FA57C4A
   ED9F0A32E84AA0FAEFD0DE9E8FD6AE8F87FB03766C834C99921EB23BE79AD9D5DC
   C1DD9AD2D36132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
   1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
   FC4963D261C8A12436518206DC93344D5AD293: FullTrust
   1.1.2. StrongName - 00000000000000000000000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

**Validate:**
1. If the Event Log permission of Grant assemblies unrestricted access to all event logs
   (unrestricted="true") is assigned to a Non-default Code Group that does not use a
   Strong Name, Publisher, or Hash as the membership condition and whose assignment
   criteria has not been reviewed and approved by the ISSO then this is a finding.
5.28 APPNET0029: Registry Permission

**Description:** The Registry permission controls application access to the Windows registry.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.RegistryPermission.

Example:
7. Dev =
   <PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">  
      <IPermission class="System.Security.Permissions.RegistryPermission", mscorlib,  
      Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"  
      Unrestricted="true" /> 
   </PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -  
      0024000004800009400000060200000240000525341310004000010010007D1FA57C4A  
      ED9F0A32E84AA0FAEDD0DE9E8FD6E6AEC8F87FB03766C834C99921EB23BE79AD9D5DC  
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82  
      1C0A5E8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0  
      FC4963D261C8A12436518206DC093344D5AD293: FullTrust  
      1.1.2. StrongName - 00000000000000000000000000000000: FullTrust  
   1.2. Zone - Intranet: LocalIntranet  
      1.2.1. All code: Same site Web  
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'  
   1.3. Zone - Internet: Internet  
      1.3.1. All code: Same site Web  
   1.4. Zone - Untrusted: Nothing  
   1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

**Validate:**
1. If a Permission Set with the Registry permission of *Grant Assemblies unrestricted access to the registry (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Registry Permission</td>
<td></td>
</tr>
</tbody>
</table>

**5.29 APPNET0030: Directory Services Permission**

**Description:** The Directory Services permission controls application access to the system Directory Service resources.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.DirectoryServices.DirectoryServicesPermission.

**Example:**
7. Dev =
   
   <PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
      <IPermission class="System.DirectoryServices.DirectoryServicesPermission,
 System.DirectoryServices, Version=2.0.0.0, Culture=neutral,
 PublicKeyToken=b03f5f7f11d50a3a" version="1" />
   </PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

**Example:**

**Level = Machine**

**Code Groups:**

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust
      1.1.1. StrongName -
      0024000004800000940000000602000002400005253413100040000010001007D1FA57C4A
      ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EF8F1645C40C93C1AB9285D62C8AA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
  1.1.2. StrongName - 00000000000000000000000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
   1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:
1. If the Directory Services permission of Grant assemblies unrestricted access to all directory service paths (unrestricted="true") is assigned to a non-default Code Group that does not use a Strong Name as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the Directory Services permission of Write() or Browse() is assigned to a non-default Code Groups that does not use a Strong Name as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

<table>
<thead>
<tr>
<th>Category</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-CSP</th>
</tr>
</thead>
</table>

5.30 APPNET0031: No Strong Name Verification

**Description:** The Strong Name Membership Condition establishes the requirement for all code defined in the group to be configured with a Strong Name. Strong Name verification should not be omitted in a production environment.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use regedit to review the windows registry for assemblies omitting strong name verification in the registry entry HKLM\Software\Microsoft\StrongName\Verification. There should be no assemblies list under this registry key.
UNCLASSIFIED

UNCLASSIFIED

.NET Framework Security Checklist V1R3
22 April 2016
Developed by DISA for the DoD

Figure 5-1: Sample Finding Example

Validate:
1. If any assemblies are listed as omitting Strong Name verification in a production environment then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>No Strong Name Verification</td>
<td></td>
</tr>
</tbody>
</table>

5.31 APPNET0032: First Match Code Groups

**Description:** The First Match Code Group is used to control the depth to which a branch of the code group tree is traversed when assigning membership to assemblies.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to list the code groups and for the First Match Code Group type.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON
Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust (LevelFinal)
      1.1.1. StrongName - 00240000048000009400000024000002534131000400000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AE8C8F87FB03766C834C99921EB23BE79AD9D5DC81DD9AD236132102900B723CF980957FC4E177108FC607784F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFA6D3754D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. (First Match) Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. Publisher - 30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630BEE2A0C2F8170A823AE69FDEB657047FD446DEFEF1F6BA12B6ACBD81BFA7B9B595AB9A40636467CFF7C73F19B53A97CF177F6E7896EBC591DD3003C5992A266C0AD9F
   1.6. Publisher - BEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFB46FC2F5CFD0203010001: FullTrust

Success

Validate:

1. If First Match Code Groups are used and the site does not have documentation regarding their use of First Match Code Groups then this is a finding.
2. Ask the System Administrator to verify the CAS policy is loaded is not the default policy. The CLR will load the default CAS policy when the policy file is corrupted. If the security policy is not loaded this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>First Match Code Groups</td>
<td></td>
</tr>
</tbody>
</table>
5.32 APPNET0033: File Code Groups, Net Code Groups

Description: The File Code Groups and Net Code Groups are used to establish directory access and web site connections respectively by the application.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5


See Section 4.6 of the checklist for .Net Configuration File Locations

Examples:


Validate:

1. If File or Net Code Groups are used and the site does not have documentation regarding their use then this is a finding.
2. Ask the System Administrator to verify the CAS policy is loaded is not the default policy. The CLR will load the default CAS policy when the policy file is corrupted. If the security policy is not loaded this is a finding.

| Category:          | CAT II       | Level: Gold MCL: 1-CSP;2-CSP;3-
|--------------------|--------------|----------------------------------|
| SDID:              | File Code Groups, Net Code groups | IA Control: DCSL-1

5.33 APPNET0035: Level Final Code Group Attribute

Description: The Level Final Code Group Attribute prevents permission sets farther down in the Code Group hierarchy from being applied to the assembly.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to list the code groups and and the Level Final code group attribute.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON Policy change prompt is ON

Level = Machine
Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust *(LevelFinal)*
      1.1.1. StrongName -
          0024000000480000094000000060200000024000052534131000400000100010007D1FA57C4A
          ED9F0A32E84A0FAEFDD0DE9E8FD64E8C88F7B03766C834C99921EB23BE79AD9D5DC
          C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
          1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
          FC4963D261C8A1243651820BDC9334AD5AD293: FullTrust
      1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. (First Match) Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. Publisher -
       30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630
       BEE2A0C2F8170A823AE69FDEB65704D7FD446DEEF1F6BA12B6ACBD1BFA7B9B595
       AB9A40636467CFF7C73F198B53A9A7CF177F655896EBC591DD300C5992A266C0AD9F
       BE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD020301
       0001: FullTrust

Success

**Validate:**

1. If Level Final Code Groups are used and the site does not have documentation regarding their use then this is a finding.

<table>
<thead>
<tr>
<th>Category</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID : Level Final Code Group Attribute</td>
<td></td>
<td>IA Control: DCSL-1</td>
</tr>
</tbody>
</table>

### 5.34 APPNET0041: Zone Membership Condition

**Description:** The Zone Membership Condition determines policy level based on the URL zone of the application origin.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to list the code groups and the zone membership condition.
Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
   1.1. **Zone** - MyComputer: FullTrust (LevelFinal)
      1.1.1. StrongName -
      00240000480000094000000062000002400005253413100040000010001007D1FA57C4A
      ED9F0A32E84A0FAFEFD0DE9E8FD6AE8FC834C99921EB23BE79AD9D5DC
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
      1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63745D6F2DE5F17E5EAF0
      FC4963D261C8A12436518206DC09334D5AD293: FullTrust
      1.1.2. StrongName - 00000000000000000000000000000000: FullTrust
   1.2. **Zone** - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
      1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
   1.3. **Zone** - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. **Zone** - Untrusted: Nothing
   1.5. (First Match) **Zone** - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. Publisher -
      30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630
      BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACDB1BFA7B9B595
      AB9A40636467CFF7C73F198B53A97CF177F6E7896EBC591DD3003C5992A266C0AD9F
      BEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5C00D020301
      0001: FullTrust

Success

Validate:
1. If a **Zone** membership condition is used for a non-default code group this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Zone Membership Condition</td>
<td></td>
</tr>
</tbody>
</table>
5.35  APPNET0045: Administering CAS Policy

**Description:** The use of the CAS policy can be enabled or disabled on the system.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to list the code groups for the enterprise policy of Search for the occurrence of Security is OFF.

Example:
C:\WINDOWS\Microsoft.NET\Framework\v1.1.43\caspol -en -lg Microsoft (R) .NET Framework CasPol 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

**Level =** Enterprise

Code Groups:

1. All code: FullTrust

Success

**Validate:**

1. If CAS Policy has been disabled then this is a finding.

| Category:          | CAT I            | Level: Gold MCL: 1-CSP;2-CSP;3-
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Administering CAS Policy</td>
<td></td>
</tr>
</tbody>
</table>

5.36  APPNET0046: Software Publishing Certificate

**Description:** The Windows system may be configured to allow use of certificates. These certificates must be validated correctly.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use regedit as in Section 4.8 of the checklist to determine the software publishing state value for all users of the system.
Validate:

1. If the *Software Publishing State* is not set to hexadecimal 23C00, then this is a finding.

<table>
<thead>
<tr>
<th>Category</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP; 2-CSP; 3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Software Publishing Certificate</td>
<td></td>
</tr>
</tbody>
</table>

5.37 APPNET0048: Publisher Membership Condition

**Description:** The Publisher Member Condition requires member code to be certified using certificates originating from a trusted source.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to list the code groups and the publisher membership condition.

Example:

Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

**Level = Machine**

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust (LevelFinal)
      1.1.1. StrongName -
      0024000000480000094000000620000002400005253413100004000000100010007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
      C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
      1.1.2. StrongName - 0000000000000000400000000000: FullTrust
   1.2. Zone - Intranet: LocalIntranet
      1.2.1. All code: Same site Web
   1.3. Zone - Internet: Internet
      1.3.1. All code: Same site Web
   1.4. Zone - Untrusted: Nothing
   1.5. (First Match) Zone - Trusted: Internet
      1.5.1. All code: Same site Web
   1.6. Publisher -
Success

Validate:

1. If the Publisher Membership Condition is used on a Non-default Code Group and the use of that Publisher's certificate is not documented and approved by the ISSO then this is a finding.

| Category: | CAT II | Level: Gold MCL: 1-CSP;2-CSP;3-
| SDID: | Publisher Membership Condition |

5.38 APPNET0052: Strong Name Membership Condition

Description: The Strong Name Membership condition requires that member assemblies be defined with Strong Names.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Use caspol to list the code groups and the strong name membership condition.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
   1.1. Zone - MyComputer: FullTrust (LevelFinal)
      1.1.1. StrongName -
         00240000004800000940000006020000024000005253413100040000100100071DFA57C4A
         ED9F0A32E84AA0FAEFD0DE9E8FD6AECE88F87FB03766C834C99921EB23BE79AD9D5DC
         C1D9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
         1C0A5EFE8F1645C4C0C93C1AB99285D622CAAA652C1DFAD63D745D6F2DE5F17E5EAF0
UNCLASSIFIED
.NET Framework Security Checklist V1R3
22 April 2016
Developed by DISA for the DoD

FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. **StrongName** - 00000000000000000400000000000000: FullTrust

1.2. Zone - Intranet: LocalIntranet
   1.2.1. All code: Same site Web
   1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet
   1.3.1. All code: Same site Web

1.4. Zone - Untrusted: Nothing

1.5. (First Match) Zone - Trusted: Internet
   1.5.1. All code: Same site Web

1.6. Publisher -
   30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630
   BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACBD81BFA7B9B595
   AB9A40636467CFF7C73F198B53A9A7CF177F6E7896BEC591DD3003C5992A266C0AD9F
   BEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD020301
   0001: FullTrust

Success

**Validate:**
1. If a Strong Name membership condition is assigned to a non-default Code Group ensure
   the private key is adequately protected by the software developer. Ask the System
   Administrator how the private keys are protected. If the private key is not adequately
   protected then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Strong Name Membership Condition</td>
<td></td>
</tr>
</tbody>
</table>

5.39 **APPNET0054: Administering CAS Policy for Group Names**

**Description:** The use of duplicate code group names within a level of the CAS policy can lead to
mis-assignment of permissions.

**Applies to:** Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Review the caspol.exe listing for all code groups. Review all code group names. The code group
names follow a sequential number at the far left of the file or screen. Code attributes for the code
group are numbered and indented below each code group name.

**Validate:**
1. If non-unique Code Group names are used within a CAS security policy level (Enterprise,
   Machine, User) then this is a finding.
5.40 APPNET0055: Administering CAS Policy and Policy Configuration File Backups

Description: CAS Policy and CAS Policy Configuration files are required for a complete system baseline and disaster recovery event.

Applies to Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Validate:
1. Ask the System Administrator if all CAS policy and policy configuration files are included in the system backup. If they are not then this is a finding.
2. Ask the System Administrator if the policy and configuration files are backed up prior to migration, deployment, and reconfiguration. If they are not then this is a finding.

5.41 APPNET0060: Remoting Services Authentication and Encryption

Description: The `typefilterlevel="Full"` attribute allows unfiltered code to access system resources.

Applies to: Versions 1.0, 1.1, 2.0, 3.0 & 3.5

Check for the CAS configuration files for `typefilterlevel="Full"` This allows references to custom client objects to be passed as parameters.

Verify .Net Configuration Files for Enterprise, Machine and User configurations for all installed versions of the .Net Framework. Search for any entries in the configuration file where `typeFilterLevel="Full"` Ask the System Administrator what encryption and authentication methods are in place for the remoting channels.

See Section 4.6 of the checklist for .Net Configuration File Locations

Example:
```
<application name="remoteserver">
  <service>
    <activated type="sample.my.object, myobjects"/>
  </service>
  <channels>
```
<channel ref="http server" port="80"/>
</channels>
</application>

<serverProviders>
<provider ref="wsdl" />
<formatter ref="soap" typeFilterLevel="Full" />
<formatter ref="binary" typeFilterLevel="Full" />
</serverProviders>

Validate:
1. In Version 1.0 of the .NET Framework if authentication and encryption are not used for all remoting channels then this is a finding.
2. In Version 1.1 or above of the .NET Framework if authentication and encryption are not used for all remoting channels when the typefilterlevel="Full", then this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID :</td>
<td>Remoting Services Authentication and Encryption</td>
<td></td>
</tr>
</tbody>
</table>

**5.42 APPNET0061: Unsupported .Net Framework Versions**

**Description:** Verify the installed .Net Frameworks are still supported by Microsoft.

**Applies to:** All Versions

Determine which versions of the .NET Framework are installed by opening the directory %systemroot%\Microsoft.NET\Framework. The following folders contain the released versions of the .NET Framework:

- v3.5
- v3.0
- v2.0.50727
- v1.1.4322
- v1.0.3705

Search for all the Mscorlib.dll files in %systemroot%\Microsoft.NET\Framework.. Click on each of the files and view properties and click Version tab to determine the version installed. If there is no Mscorlib.dll, there is no installed version of .Net Framework in that directory.

More specific information on determining versions of .Net Framework installed can be found at the following link. [http://support.microsoft.com/kb/318785](http://support.microsoft.com/kb/318785)


<table>
<thead>
<tr>
<th>.NET Framework 1.0</th>
<th>7/14/2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>.NET Framework 1.1</td>
<td>10/8/2013</td>
</tr>
<tr>
<td>.NET Framework 2.0</td>
<td>4/12/2016</td>
</tr>
</tbody>
</table>
Validate:

1. If the any versions of the .Net Framework are installed and extended support is no longer available this is a finding.

<table>
<thead>
<tr>
<th>Category:</th>
<th>CAT II</th>
<th>Level: Gold MCL: 1-CSP;2-CSP;3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDID:</td>
<td>Unsupported .Net Framework Versions</td>
<td></td>
</tr>
<tr>
<td>Reference:</td>
<td>DoDI 8500.2 IA Controls</td>
<td>IA Control: COMS-1 COMS-2</td>
</tr>
</tbody>
</table>